(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
3 May 2001 (03.05.2001)

PCT

(10) International Publication Number
WO 01/31422 A2

(51) International Patent Classification[7]: G06F 1/00

(21) International Application Number: PCT/ZA00/00192

(22) International Filing Date: 19 October 2000 (19.10.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/161,047 25 October 1999 (25.10.1999) US

(71) Applicant and
(72) Inventor: VON WILLICH, Manfred [ZA/ZA]; 41, 1st Avenue West, Parkhurst, 2193 Johannesburg (ZA).

(74) Agents: DUNLOP, Alan, J., S. et al.; Hahn & Hahn Inc., 222 Richard Street, Hatfield, 0083 Pretoria (ZA).

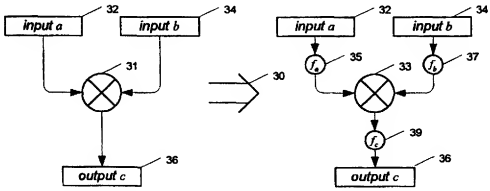
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AT (utility model), AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, CZ (utility model), DE, DE (utility model), DK, DK (utility model), DM, DZ, EE, EE (utility model), ES, FI, FI (utility model), GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *Without international search report and to be republished upon receipt of that report.*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: METHOD FOR MAKING DATA PROCESSING RESISTANT TO EXTRACTION OF DATA BY ANALYSIS OF UNINTENDED SIDE-CHANNEL SIGNALS



(57) Abstract: The invention provides a method of processing of and storing data to reduce the risk of unauthorized access to the data, especially through side-channel observations. The method includes the steps of designing of algorithms, particularly ciphers, for maximum benefit from this technique, modifying the algorithm implementation to operate on mapped data, initially mapping of data, especially cryptographic keys, for storage, changing the data mapping from a prior data mapping by use of a secondary mapping, mapping incoming data for input to the modified algorithm implementation, and mapping data output from the modified algorithm for further use. The method results in enhanced secrecy of the original data and the mapping on the data. The data mapping and the secondary data mapping may be in the form of a lookup-table, an algorithm with mapping selection data, or the like. The data mapping may be implemented as cascaded mappings to further reduce the risk of unauthorized access.

WO 01/31422 A2

METHOD FOR MAKING DATA PROCESSING RESISTANT TO EXTRACTION OF DATA BY ANALYSIS OF UNINTENDED SIDE-CHANNEL SIGNALS

FIELD OF THE INVENTION

This invention relates to data security. In particular, this invention relates to reducing the risk of unauthorised access to data.

BACKGROUND OF THE INVENTION

Side-channel attacks on secret data

Cryptographic systems have traditionally been depicted with the cipher (encryption or decryption) as a metaphorical black box, in which input data (whether plaintext or ciphertext) is processed internally using a secret key and the only information to leave the black box is the intended output data.

It has been shown recently (for example, in: P. Kocher, J. Jaffe and B. Jun, Differential Power Analysis, Advances in Cryptology – Proceedings of Crypto '99, Lecture Notes in Computer Science, Vol. 1666, Springer-Verlag, 1999) that side-channel information, such as unintended radiated electromagnetic radiation or fluctuations in the power drawn by a device, may be exploited easily and effectively in an attack aimed at monitoring information being processed. This makes it much easier to extract the secret key than the traditional cryptanalysis model would lead us to believe, since a direct observation, albeit noisy, of the internal processing becomes available to the attacker.

Where a ciphering operation uses a key repetitively, the attacker can generally obtain it by observing and analysing the side-channel information during several operations, without resorting to traditional techniques of cryptanalysis. The minimum number of repeated operations that must be observed to extract the value of the key (or any repetitively used internal data) typically changes in inverse proportion to the ratio of the power of the signal he is trying to observe to the power of the noise (the signal-to-noise ratio). As an example, where a hardware modification decreases this ratio 100-fold (i.e. by 20dB), the attacker will typically need to observe in the order of 100 times as many operations to extract the key.

There may be practical and economic limits to the reduction in the signal-to-noise ratio of the side-channel, such as by shielding and addition of noise. Where a secure

processor (such as a chip-card) is left in the hands of a potential attacker, he can easily stimulate the processor repetitively by providing input data while observing the side-channel closely. Examples of chip-cards include banking and Pay-TV cards. With very little expense and time, the attacker may be able to extract the information he is interested in by using statistical techniques, herein called DPA (Differential Power Analysis). This naming is due to the most popular side-channel for monitoring chip-cards being observation of fluctuations in the power drawn by the device. The technique of DPA may alternately be applied to covert reception and analysis of radio-frequency signals radiated by a computer performing data manipulation.

With simple chip-card designs, analysis of differences in the averages of groups of several similar waveforms may allow secret data to be deduced. This is an example of a first-order DPA attack. It has been demonstrated that currently available commercial chip-cards, almost without exception, are vulnerable to such an attack with resources available to most determined individuals. With suitable algorithmic design and inclusion of randomness, it is possible to keep data secret in the face of a first-order or even a higher-order DPA attack.

The order of a DPA attack may be defined as the minimum number of intermediate variables from which the any of the data exposed by the attack may be obtained, where these intermediate variables are each derived from the observations by an averaging process over a large number of observations. A more intuitive (but less accurate) definition may be that it is the number of internal digital states of which direct (if noisy) side-channel observations must be made to obtain any information about the information desired by the attacker.

With more sophisticated processing of the data (a so-called high-order DPA attack) and a larger number of observations, it will remain possible in principle to determine with some degree of confidence any secret data being processed, although the necessary number of observations can be made prohibitively large.

The objective of the techniques of the invention presented here is to reduce the amount of useful information an attacker may obtain from the side-channel signal and to increase the minimum sophistication and complexity of a successful attack. The techniques include defence against first- and higher-order attacks.

In general, a design objective in secure devices with regard to data secrecy would be to keep the amount of leaked information about secret data during the life of the secret below acceptable limits. This may be achieved through cryptographic mechanisms of making the process of combining small quantities of leaked information into a useable whole computationally intractable. It may also be achieved by limiting the rate of leakage of information so that the cumulative leakage throughout the life of the secret of information (defined in an information-theoretic sense) about the secret is acceptably low, as is the objective of this invention.

Mathematical background

A set of data (e.g. bits) may be made mapped onto another set of data in such a way that the original set of data remains entirely unknown to an observer despite the second set of data being known to the observer. The original data (the first set) may be reconstructed from the mapped data (the second set) when the selection of mapping is known. To retain data secrecy, the selection of mapping must be unknown to the observer and the mapping must be selected randomly for every new set of data in such a way that every possible original data set will be mapped to every possible mapped representation with equal probability. This principle is exploited by this invention.

Operators (for combining one or more operands into a result) are used as building blocks in cipher design. Examples of such operators include a lookup table – a unary operator – modular addition or subtraction, word-wide bit-for-bit exclusive-or, and modulo-p multiplication (over the set of values 1 to $p-1$, p being a prime number) – the latter all being binary operators. The well-known IDEA cipher (designed by Xuejia Lai and James Massey) uses three such binary operators, and the well-known DES cipher uses lookup tables, the bit-for-bit exclusive-or operator and bit-permutations.

In general, a separately and arbitrarily selected one-to-one mapping may be applied to each of the inputs and to the output of any operator. An equivalent operator may then be defined that generates the correct mapped output from the mapped input values for every selected mapping. For any given operator, there may exist a set of such mappings such that this equivalent operator is identical to the original operator and which satisfies requirements for not revealing information about the original data. The principle, including the restriction to an identical operator, is often termed blinding, although the extent of the range of mappings possible for typical operators is seldom realised.

As an example, mapping of the modulo addition operation – $x + y \equiv z \pmod m$ – under the constraint that the operator remains unchanged permits the family of mappings from (x, y, z) to (x_i, y_i, z_i) where $x_i \equiv a_i x + b_i \pmod m$, $y_i \equiv a_i y + c_i \pmod m$ and $z_i \equiv a_i z + b_i + c_i \pmod m$, where a_i is any number that is mutually prime with m, and b_i and c_i are any numbers. Where m is a power of 2 (i.e. of the form $m = 2^n$) there are $m/2$ possible values for a_i and m possible values for each of b_i and c_i. Many field operations (such as addition, multiplication and exponentiation) will exhibit similar properties.

The operation of a word-wide exclusive-or of bits (which we consider here as addition of two vectors of n components over the field Z_2, addition and multiplication are equivalent to binary "exclusive-or" and "and" operations respectively, and we use lower case to indicate a vector and upper case to indicate a matrix) – $x + y = z$ – has a larger selection of data mappings than modulo-2^n addition has under the constraint that the operator is to remain unchanged. These have the form $x_i = A_i x + b_i$, $y_i = A_i y + c_i$ and $z_i = A_i z + b_i + c_i$. A_i may be any of $\prod_{k=0}^{n-1}\left(2^n - 2^k\right)$ matrices – those with an inverse – and b_i and c_i may each have any of 2^n values, giving $2^n \cdot \prod_{k=0}^{n-1}\left(2^n - 2^k\right)$ distinct mappings for each value (ignoring constraints implied by the shared matrix A_i). Where $n = 8$ bits, there are approximately $2^{70.2}$ such mappings.

The size of the set of mappings available for the exclusive-or operation can significantly reduce the usability of the side-channel signal, and in so doing may permit compromising of some of the requirements for secrecy. Such compromise (e.g. re-use of selection of mapping) can be useful in reducing the complexity of the final design of an algorithm while keeping the amount of information leaked to the attacker acceptably low.

Multiple mappings may be applied to the same data consecutively to make a composite mapping – e.g. $x_i = f_i(x)$, $x_{ij} = f_j(x_i)$. Although this is equivalent to a single mapping $x_k = f_k(x)$, where $f_k = f_j \circ f_i$, if arranged correctly the attacker must obtain information about multiple independent sets of data (three in the example – x_k, f_i, and f_j) before obtaining any information about the original data. This increases the order of the DPA attack (typically equal to the number of independent sets of data) and the number of observations required (typically as the power of the number of independent sets of data) before being able to extract useful information from the observations.

Unary operators (such as a lookup table or a bit-permutation) also find application in ciphers. Mappings that allow the operator to remain unchanged are restricted only when there is data loss in the operation (i.e. it is many-to-one), but may make more sense to modify the operator in these instances, for example by use of a mapping-dependent lookup table.

International publication number **WO 99/67919** to Kocher, Jaffe and Jun proposes methods and apparatuses for improving DES cryptographic protocol against external monitoring attacks by reducing the amount (and signal-to-noise ratio) of useful information leaked during processing. An improved DES implementation of the invention instead uses two 56-bit keys (K1 and K2) and two 64-bit plaintext messages (M1 and M2), each associated with a permutation (i.e., K1P, K2P and M1P, M2P) such that K1P{K1} XOR K2P{K2} equals the "standard" DES key K, and M1P{M1} XOR M2P{M2} equals the "standard" message. During operation of the device, the tables are preferably periodically updated, by introducing fresh entropy into the tables faster than information leaks out, so that attackers will not be able to obtain the table contents by analysis of measurements. The technique may be implemented in cryptographic chip-cards (smartcards), tamper resistant chips, and secure processing systems of all kinds. Where blinding is used, the relationship between the number of observations needed to extract useful information via a side-channel and the power SNR of this channel differs from that of inverse proportionality, and no indication of the understanding of this principle is given in the application. In the case of blinding as in this proposal (with or without permutation), the number of observations needed should be expected to vary inversely with the square of the power SNR (i.e. the fourth power of the magnitude SNR).

SUMMARY OF THE INVENTION

The technique of the invention provides a practical and effective modification of cryptographic and other processes, such modification being based on data secrecy through varying of the mapping of all secret and intermediate data for computation and storage. Examples of such data are cryptographic keys, stored and communicated data.

Where either the mapped data or the selected mapping (or all mappings of a composite where used) is unknown, no information about the secret data can be determined. This technique has the potential to reduce the amount of information obtainable about the original data from side-channel leakage significantly, provided the observable side-channel leakage is sufficiently low.

Secret data, most particularly cryptographic keys, are never needed in the original form (without an applied mapping) with the exception of their use in the initial mapping, and are randomly re-mapped on a per-use basis to avoid data repetition that would facilitate a DPA-attack.

An example of when this technique will have high value is in chip-cards, where DPA may in some cases provide an unauthorised party with the data in use within minutes, entirely though analysis of the leaked side-channel signals. Another potential use is in computation and storage of data on computing devices where electromagnetic radiation may compromise the secrecy of the data.

Thus, in order to lead to the benefits of the invention, there is provided a method of processing of data to reduce the risk of unauthorised access to the data, for example, by DPA, the method including the steps of:

- design of algorithms, particularly but not exclusively ciphers, for maximum benefit from this technique;
- extending the commonly known technique of data blinding to a larger set of mappings;
- modifying the algorithm implementation to operate on mapped data;
- initial mapping of data, especially cryptographic keys, for storage;
- changing of the data mapping from each prior data mapping by use of a secondary mapping;
- mapping incoming data for input to the modified algorithm implementation; and
- mapping data output from the modified algorithm for further use.

The method may include the keeping both the secret data and the selection of mapping on the data secret.

The data mapping and the secondary data mapping may be in the form of a look-up-table, an algorithm with mapping-selection data, or the like.

The methods may include composite (cascaded) but separately applied mappings to reduce the amount of information that may be obtained from a given number of observations by an attacker and to increase the lowest order of a successful DPA attack.

The mapped data and the selection of mapping may be transmitted to a remote location.

DESCRIPTION OF THE DRAWINGS

The invention may be better understood with reference to the following explanations, the non-limiting examples and the accompanying drawings.

Figure 1 shows, in schematic representation, a prior art cryptographic operation;

Figure 2 shows, in schematic representation, side channel information leakage in the operation of Figure 1;

Figure 3 shows, in schematic representation, replacement of a two-input operation with a data-mapped equivalent;

Figure 4 shows, in schematic representation, combining of consecutive mappings;

Figure 5 shows, in schematic representation, replacement of a cipher by its modified equivalent;

Figure 6 shows, in schematic representation, initial mapping of the key for storage;

Figure 7 shows, in schematic representation, the iterative mapping of a key;

Figure 8 shows, in schematic representation, a simplistic cipher illustrating the mapping process; and

Figure 9 illustrates aspects of Example 3: Making the DES cipher resistant to both 1st- and 2nd-order DPA attacks.

In Figure 1, reference numeral 10 generally indicates a traditional "black box" cryptographic operation. In operation 10, the input data 12 is transformed using a key 14 to an output 16.

In Figure 2, reference numeral 20 generally indicates a traditional cryptographic operation, such as that shown in Figure 1, further indicating side-channel leakage. The operation 20 includes inputting of data 22, the transformation by key 24 to output data 26 and leakage of signal 28.

In Figure 3, reference numeral 30 generally indicates the process of replacement of a two-input operation with a data-blinding equivalent. In the operation 30, a standard two-input operation is represented with inputs 32 and 34 being operated upon by operator 31 to produce an output 36. The data blinding operation again takes inputs 32 and 34, that are then mapped by mappings 35 and 37 before being operated upon by operator 33. The combined output is then mapped by output mapping 39 to provide the hidden output data 36.

In Figure 4, reference numeral 40 generally indicates the process of combining of

consecutive mappings of Figure 3 from cascaded operations. Operators 41 and 47 correspond to two distinct instances of operation 33 of Figure 3. Mapping 43 corresponds to the output mapping 39 in relation to operator 41, and mapping 45 corresponds to an input mapping (such as 35 or 37) in relation to operator 47. Mapping 49 ($f_{c.d}$) is a single composite mapping derived from 43 and 45 that does not generate any data correlated to the original data even as an intermediate value.

In Figure 5, reference numeral 50 generally indicates the replacement of a cipher by its modified equivalent (as an intermediate step of deriving a final implementation of the invention). It can be seen that in the unmodified cryptographic operation, input data 52 is acted upon by ciphering operation 53 using key 51, rendering an output 54. In the modified equivalent, the input data 52 is transformed by transformation 56 into a mapped form prior to being acted upon by modified cipher 57 using a key in mapped form, rendering a mapped output from which the original output 54 may be derived using transformation 58.

In Figure 6, reference numeral 60 indicates the process of making an unpredictable selection of a mapping. The unmapped key 62 is mapped 63 according to the selection made and stored 64. The mapping selection is stored 68 for use with the mapped key.

In Figure 7, reference numeral 70 indicates the process of making an unpredictable selection of a secondary mapping. The previously mapped key 72 is further mapped 73 by use of the selected secondary mapping and stored 74, typically replacing 72. The previously stored mapping selection 76 is processed with knowledge of the selection secondary mapping selection to yield the mapping selection applicable to 74, and this is stored 78, typically replacing 76.

In Figure 8, reference numeral 80 generally indicates the process of replacing of an algorithm with an algorithm that operates on mapped data. The cipher 83 operates on an input text 81 and key 82 to yield an output text block 84. In the replacement, the input text is mapped 85 using one or more suitable mappings. Optionally, the initial key 82 is similarly mapped 86 to yield a mapped key 89. Alternatively, 89 may be provided from the output of a decryption operation already in mapped form. 86 further refers to repeated changing of the mapping applied to the key. The modified cipher 87 operates on the mapped data, and its mapped output is optionally operated upon 88 by a mapping operation to yield the same data 84 as would have been yielded by the unmodified cipher. Alternately, the output of 87 may be used directly with the mapping selection data in similarly modified algorithms at the equivalent 85 and 86 to avoid occurrence of the unmapped form of the data.

In Figure 9, reference numeral 90 generally indicates the process of replacing bit-permutations with the manipulation of mapped data and mapping selection data for independently applied mappings for each mapped data bit. Reference numeral 91 similarly indicates the replacement of duplication of a data bit without the introduction of differentiation between the mappings, but with the caveat that care must be applied with regard to recombination of such data introducing unwanted cancellation of unpredictability. Reference numeral 92 represents the same replacement operation, except that unpredictable information 95 is introduced to avoid the caveat mentioned for 91. Reference numeral 93 similarly indicates the replacement of an exclusive-or operation. Reference numeral 94 indicates the replacement of a DES S-function lookup table (having six input bits and four output bits) with a pre-calculated lookup table using mapped values. In the pre-calculation, unpredictable data 96 and all possible input values 97 are combined with the original table to generate all the mapped input-output combinations 98 for writing into the mapped lookup table 99. This pre-calculation may be done for every use or for multiple uses of the table according to design choice. This lookup table 99 is then used in conjunction with adequately isolated re-mapping operations (exclusive or) to operate on mapped data. No two vectors of bits in the diagram can be used to reconstruct the original data. To obtain sufficient isolation, it may be necessary to introduce delays into signal paths (such as through the use of clocked latches between exclusive-or operations).

DETAILED DESCRIPTION OF THE INVENTION

Cipher design

Care should be exercised in choice of cipher algorithm. Suitable cipher design can result in the next step (cipher modification) adding very little processing overhead to the cipher. Choosing the set of operations that are used in the cipher is important to minimise complexity and maximise data secrecy in the face of a side-channel attack. Understanding of the following aspects of the technique is essential during the design.

Reuse of a mapping with different sets of data within an algorithm will introduce potential weaknesses to an attack, but where such a weakness is not too severe (such as where resistance to only a first-order DPA attack is required), this may lead to significant savings in added computation. This must be borne in mind in the algorithm design.

Cipher modification

A fresh selection of mapping may be used for each data value (including the output of every operation) throughout the cipher, or else the mapping may be left unchanged between two operations. The latter is typically not possible when the two operations are unrelated, but when possible may be useful in keeping complexity low. Care must be exercised that the mapping associated with all intermediate computational values adheres to the hiding requirements (for example, where two values that have the same mapping applied are combined through an exclusive-or operator, an original output of zero will always be mapped the value zero).

Every operation is substituted with one that performs the equivalent operation with all values being mapped, as illustrated in Figure 3. The output mapping 39 (f_c) is determined by the input mappings 35, 37 (f_a and f_b) and any changes to the core operation. For example, where input mappings are composed of adding separate randomly selected values to each of the inputs of an addition operation, the output mapping would be composed of subtracting the sum of the random values from the output, assuming the core addition operation is kept identical.

The original values 32, 34 and 36 (a, b and c) still occur in Figure 3, but do not occur after the next step has been applied. The operation performed on the mapped values will normally be chosen to be the same operation as before if finding a suitable alternative is impractical (e.g. for addition), but may be different when replacement is reasonable (e.g. for an arbitrary lookup table).

The next step is to combine consecutive mappings 43 and 45 (f_c and f_d) from cascaded operations 41 and 47 into a single mapping 49 ($f_{c,d}$), as illustrated in Figure 4. This mapping must not, even as an intermediate calculation value, derive the original data or any data correlated to the original data. This will in general be achieved when the mapping 49 is constructed only from information that cannot be used to derive information about the original data from the mapped value. Occurrence of correlated data would provide a primary target for a DPA attack. For example, if the two mappings 43 and 45 are modulo addition of separate random values, the mapping 49 will be addition of the sum of these values, from which no information about the individual mapping selections may be deduced. Where the adjacent operations are related, this mapping may be simplified. Where the selections of consecutive mappings 43 and 45 are correlated (i.e. the selection of one influences the selection of the other), the composite mapping may be somewhat simpler or may even

become the identity operation (and hence be omitted).

Where the cascaded operators 41 and 47 are unrelated, a complex operation may be necessary as the implementation of mapping 49 ($f_{c,d}$). If necessary, it may be implemented by use of a lookup table or another operation. If one of the adjacent operations is a lookup table, the resulting cascaded lookup tables may be combined into one lookup table. After this step, aside from the input data, key-data and output data, the data in all computations are kept secret by the mappings. These external mappings are treated separately in the next steps.

With careful choice of cipher design and restrictions on selection of mappings, the complexity of the modified cipher need not be much greater than that of the original cipher, disregarding the mapping selection, manipulation and mapping external to the modified cipher 57. Computation relating to the mapping used in each operation may be kept to minimum. The resulting mathematically equivalent cipher is shown in Figure 5.

Initial storage of keys

In Figure 5, the original key, input data and output data are still shown as occurring without an applied mapping, and may still the be target of a DPA attack when these are accessed by an operation, in particular for the mapping process. The cryptographic key must be stored only in a mapped form, where the selection of mapping has the required randomness. Additionally, the information encoding the selection of mapping must be stored. This initial storage is only needed when the initial or master keys are downloaded (typically in a protected environment), and never for keys downloaded in encrypted messages (see *Cipher output data mapping*). This may be expressed as initially storing the key k with an applied mapping $k_0 = f_0(k)$, as well as information identifying the selection of mapping, f_0. The family of mappings will most commonly be chosen in relation to the operators used in the cipher in which the key is used to avoid unnecessary re-mapping.

Per-use key mapping

Even stored with an applied mapping as in *Initial storage of keys*, repeated accesses would allow both the secret data and the mapping information to be reconstructed through first-order DPA techniques (e.g. through analysing averages of groups of observed traces). Therefore, prior to each use of a cryptographic key, the mapping should be replaced with a fresh, randomly selected mapping subject to the constraints imposed by the design. The

original value of the key must not be computed, even as a temporary variable, in this process. This leads to deriving of the values in the form $k_i = g_i(k_{i-1})$ and $f_i = g_i \circ f_{i-1}$. By the latter is meant deriving f_i such that $f_i(q) = g_i(f_{i-1}(q))$ for any q. The values k_i and f_i will replace the stored values k_{i-1} and f_{i-1}. These values will remain related by the identity $k_i = f_i(k)$.

Cipher input data mapping

The input data 52 (x in Figure 5) is first mapped using the mapping selected for those inputs. This is analogous to the initial mapping of the key (under *Initial storage of keys*), but may occur with all data to be processed, such as received ciphertext to be decrypted or plaintext to be encrypted for transmission. Where sensitive data (e.g. keys) are to be encrypted, they must already be stored in mapped form and have a mapping substitution performed where appropriate (as in *Per-use key mapping*).

Cipher output data mapping

The output may be mapped to its original value where its secrecy is not critical (e.g. where ciphertext has been generated for transmission). Where this data must remain secret (e.g. transmitted cryptographic keys), they and the mapping selection information should be stored without being mapped back to the original form. Thus, the initial mapping of the key mentioned above does not occur with received and decrypted keys. This makes the process of downloading keys resistant to DPA.

Example 1: Making an "exclusive-or" based cipher DPA-resistant

In this example, a simplistic cipher is constructed entirely from modulo-2 addition – exclusive-or – of octets (vectors of eight bits each) and a single lookup table that produces an 8-bit output value for each 8-bit input value. Due to the simplistic nature of the cipher, only a single set of data may be ciphered securely for the use of the key (as in a Vernam cipher or one-time pad), but repeated ciphering of the same data is provided with first-order DPA-resistance. The per-use key mapping has not been shown, and is necessary for DPA-resistance. However, this example is intended to illustrate cipher design for use within a severely constrained computational environment, such as a chip-card. It uses a single lookup table substitution.

Related mappings are applied to every data octet in this example, of the form $k_{n,i} = A_i k_n + b_i$, $x_{n,i} = A_i x_n + c_i$ and $y_{n,i} = A_i y_n + d_i$. The subscripts n and i refer respectively to

the selection of the octet within each data set and the cipher use count. A_i is a randomly selected non-singular 8-by-8 matrix of bits and each b_i, c_i and d_i is a randomly selected octet.

In Figure 8, these operations have been combined to illustrate the example. A typical cryptographic cipher (encryption or decryption) would use many more operations and the data sizes of k, x and y would typically each be at least 64 bits. Each arrow represents the flow of one octet. The diagram shows equivalent operations with mapping of the data. Initial and incremental mapping of the key (described in *Per-use key mapping*) are both shown under key mapping.

The initially mapped key $k_{n,0} = A_0 k_n + b_0$ and the mapping $f_0 = (A_0, b_0)$ are stored.

Preferably prior to any use of the key, a fresh mapping is performed by selecting new G_i and h_i. We replace $k_{n,i-1}$ by $k_{n,i} = G_i k_{n,i-1} + h_i$, A_{i-1} by $A_i = G_i A_{i-1}$ and b_{i-1} by $b_i = G_i b_{i-1} + h_i$.

Every lookup-table s is replaced by its equivalent s_i for operation on mapped values, defined by $s_i(z) = A_i s(A_{i-1}(z + b_i + c_i)) + d_i$. Map the input data octets x_n using the related mapping, $x_{n,i} = A_i x_n + c_i$. Cipher the mapped input using the original cipher except for the substituted lookup table. Aside from the per-key mapping, the substituted lookup table, the initial mapping and final mapping, there is no change to the computation involved in the cipher.

Finally, where the output y is to remain secret, such as with a key, use y_i, A_i and d_i are used instead of y. If it is to be mapped into its original state, this may be expressed as $y_n = A_{i-1} y_{n,i} + d_i$.

A crucial observation to be made is that due to the large number ($2^{70.2}$) of possible mappings, the same mapping can be used for effective secrecy of more than one octet of data. This allows the modified cipher to remain simple. A simpler mapping may not keep multiple bytes adequately secure against DPA. Simplification on the basis of re-use of the same mapping should be minimised, and where feasible, the mappings selected for distinct data sets should be independently selected.

Since the mapping (A_i, b_i) and the mapped data d_i are changed on every use, the processed data (including the key) is not correlated with the original data. Only a function of several bits of data and the mapping is correlated to the original data. Each bit of the original data can be expressed as a function of 17 bits being processed.

This example, applied to a cryptographically strong cipher, may be used effectively in chip cards available today, including those that use 8-bit processors and modest quantities of storage space.

Example 2: Making the IDEA cipher DPA-resistant

This example illustrates the use of this concept as applied to a well-known cipher that was designed without any attempt at resistance to DPA.

The IDEA cipher was deliberately composed of three mutually incompatible operators based on primitives readily available on most general-purpose computers – binary exclusive-or, addition and multiplication of 16-bit quantities. To make this cipher DPA-resistant, due to the incompatibility of the operators a lookup table is introduced in every data path in order to map the mapped value from one operator to the next.

Each exclusive-or may have a mapping as with the example above, except that the vector size is increased to 16 bits. In the above context, by "each" is meant that the random mapping is not constrained to be the same throughout the cipher, and can be independently selected wherever a re-mapping is performed.

The addition operator has less freedom of selection of mapping than the exclusive-or operator. The multiplication operator has mapping selection freedom similar to that of the addition operator. Mappings must be randomly selected from a suitable set, the key and data must be mapped accordingly, the lookup tables must be generated and the cipher must be executed.

The overhead here is a number of lookup tables of 65536 16-bit words each, storage of information identifying the mappings applied to the key, and the processing overhead of about twice as many lookups as there are operations performed.

In the typical modern-day personal computer, these resources are readily available. As this example shows, many existing applications may easily be secured against most DPA attacks using this technique. DPA using covertly intercepted electromagnetic radiation from a computer executing a cryptographic process is readily made impractical using this approach.

It must be borne in mind that where a large amount of data is to be processed, the data mapping should be updated at intervals in the process.

Example 3: Making the DES cipher DPA-resistant

The Data Encryption Standard (DES) cipher is widely used, and although its 56-bit key length makes it vulnerable to exhaustive-search attacks, still finds wide application. It is also used in more secure variants such as Triple Data Encryption Algorithm (TDEA, more commonly known as triple-DES) and DESX (a cipher derived from DES). It is thus appropriate to consider the application of this invention to DES.

DES was not designed with DPA in mind. As is often the case, measures that are intended to increase the cryptographic strength in have reduced the compatibility of mappings that may be economically used for subsequent operations. Three significant operations are used in DES – modulo-2 addition (exclusive-or), expansion (much like a permutation, except that some or all of the input bits are duplicated) and eight 6-to-4-bit lookup tables (termed S or selection functions). Shifts, bit-permutations (re-ordering) and register interchanges are ignored in this discussion, since the mapping selections applied to each bit are simply tracked (assuming the signals are kept isolated) without having to treat these as distinct operations with the chosen mapping strategy. The replacement of unmodified bit-movements by modified bit movements including tracking of the mapping selection is illustrated in 90.

Although the specific permutations, expansions and exclusive-or operations used allow a large set of mappings on the data (including the key), any mapping involving several bits must inherently be re-mapped to allow use of only six bits at a time as input to each S-function. To consider the eight S-functions collectively as a single entity for this purpose would be prohibitive. For the purpose of simplicity of this example, mappings involving more than one bit will not be considered here. This does not imply that more complex mapping with re-mapping after nearly every operation is necessarily complex.

The mapping that will be considered here involves a separate selection for every bit being processed in the algorithm. To reduce the need for fresh random data, a compromise may be made that allows the selections to be correlated, although special care is required here to ensure that the order of DPA-resistance is not reduced below the desired order. Implementation of this compromise will not be included in the example.

Assume for this example that we desire 2nd-order DPA-resistance. To achieve this, we will use the principle that the number of independent digital quantities needed before being able to reconstruct any information about the original data must be one higher, i.e. 3. Due to the fact that digital signals interact in several unexpected ways, signals cannot be assumed to be independent unless they are suitably isolated.

Isolation of signals in a general-purpose processor is often far less than the functional description would imply. For example, loading a value into a register such as an accumulator may result in hidden operations for potential future use, such as determination of whether the value is zero. Erasure of data from a circuit followed by a time-interval before loading of further data will normally provide sufficient isolation, even though subtle interactions will occur (such as data-dependent heating or ion migration). Interaction between data values in RAM words that are not accessed directly may still be visible during other accesses due to the implementation of the addressing logic. Here, we assume an implementation in hardware with data-storage registers with suitable properties. The first of these properties is that once data has been erased from a circuit and a suitable interval (e.g. one clock cycle) has elapsed, there will be no interaction with subsequent data on that circuit. The second property is that interaction between data in separate circuits is negligible, although a more conservative form of this property is that interaction between data in separate circuits is suitably isolated provided no data-related signal transitions occur at a similar time both circuits. The point to note is that different data bits processed simultaneously generally cannot be assumed to be isolated and hence should not be treated as independent.

We assume that all input data for the algorithm is provided in mapped from (a single bit of mapped data is represented as a single bit), with two independently applied mappings for each bit, with each mapping having a single independent an unpredictable bit of mapping selection information. Each mapping is chosen from a set of two. The first mapping of the set leaves the data bit unchanged, and the second interchanges the two possible values. It will be seen that were the three bits associated with the original bit combined using an exclusive-or operation, the original bit would result. Omitting any single bit of the three

would provide secrecy of the original data (assuming each mapping selection is unpredictable, each case is equally probable, and no form of correlation exists between selections).

In modifying the DES cipher, any mapped bit is passed through any permutation as before, tracking the associated mapping bits (90). The same occurs for the expansions, except that where bit-duplication occurs (including where all bits are duplicated), the resulting duplicates should be made independent unless further analysis indicates this is not necessary (in which case the modification is as in 91). Using two independently selected mappings (two fresh bits of random data being needed), composite mappings with the previous mappings can then be formed. The incoming mapped data may have the two pairs of mappings applied (using the exclusive-or operation separately to re-map data for each duplicate, plus the composite mapping selection being deduced for each of the two mapping selections for each duplicate. This involves four new unpredictable selection bits, of which one associated with each incoming mapping may be omitted with no loss of security (as in 92). The two duplicates are not correlated with each other, and can be used together in further operations without fear of subsequent combination introducing DPA weakness. The exclusive-or operations in the DES cipher under this mapping remain the same, with composite mappings being deduced for the first and second mappings applied to each bit. This determination can be tracked in hardware – when the exclusive-or of two mapped data bits is found, every selection bit is combined with the corresponding selection bit of the other data mapping using the same operation (93). The resulting three bits are then treated as the masked data bit and two applied mappings. Selections may at times be judiciously made in a correlated fashion without reducing the lowest order of a viable DPA attack, with the effect that the additional computational complexity and unpredictable data requirements may be reduced.

It should be noted in the above that some simplifications suggested in the description of the invention have already been applied in this example. In particular, the output mapping of a previous operation is made the same as that applied to the inputs of an exclusive-or operation, simplifying the re-mapping to the identity operation (allowing omitting thereof). With data expansion (92), only sufficient extra unpredictability (95) is added to ensure that no subsequent operations may reduce the DPA-resistance. Subsequent simplification based on deliberate correlation of mapping selections may be possible.

The S-functions are the only areas (within the cipher) remaining to be addressed in this example. In keeping with the nature of the mappings chosen thus far, we will restrict the choice of mapping for the purposes of this example to mappings on individual bits. There are a variety of approaches that may be adopted when introducing simplifications. The approach taken here is to re-map the input data according to fresh mappings selected for the function (lookup table) inputs (thus allowing more than one use of the table) and to combine the input mapping unpredictability with that of the lookup table output mapping. The re-mapping approach is used here when modifying the S-function lookup table (94), but it will be kept in mind that the output bits of S-functions should be unrelated to those of the inputs. We select two mappings unpredictably and independently of all other mappings for each S-function input and output bit (96), and replacement entries for the S-function table (99) are written (98) for every possible input (generated, in this example, by counter 97) prior to use. The mapped inputs are then re-mapped, the written mapped S-function table applied, and the selected output mappings are propagated combined with the input mapping for added unpredictability, although a simplification may be done. The S-function lookup table is stored in hardware registers (in all this will be 2048 register bits to implement the eight S-functions) or RAM. The values stored in this storage must be pre-calculated from the applicable mappings in a manner that preserves the desired resistance to a 2^{nd}-order DPA attack. Each resultant mapped input value is used to address the register file and the mapped value is stored in the selected four register bits.

The introduction of random bits (as with 95 in modification 92) for data expansion is normally expensive. In this instance (under the assumption that the S-function output mapping is unrelated), this can be dispensed with as in every case the duplicates are inputs to exclusive-or operations with non-correlated data. Due to multiple key-bit duplication, fresh S-function output mapping is required to ensure this non-correlation. Also to be considered is that each key bit is used on average nearly 14 times.

The calculation of a fresh set of S-function tables for every round of DES (there are 16 rounds for every application of the cipher) including obtaining two unpredictable bits for every output bit (of which there are 32 for each round) may be extremely expensive. In hardware, it may be possible to recalculate the S-function table on every round of the cipher. Due to this cost, a typical implementation would re-use these lookup tables without alteration for more than one lookup, and possibly even for more than one invocation of the cipher algorithm. This violates the previous assumption of non-correlation of the output mappings of output bits on distinct rounds, and care must be taken to determine where this will reduce the lowest order of a viable DPA attack. In addition, the strength of the side-channel signal

correlated to an internal bit increases with the amount of use, and this must be taken into account in determining whether the leakage signal is sufficiently small. In some cases, it may be necessary to retain the unpredictable re-mapping, especially of a bit of the key (which is used several times). Another restriction resulting from this simplification is that the input mapping of a bit of an S-function must be the same for every use prior to replacement of the lookup table content. Rather than restricting the mapping applied to individual bits of data (including the key), the mapping applied to the data must be replaced by the pair of mappings applicable to the inputs to the S-function. This should be done in two steps (using two separated exclusive-or operations), each applying the composite of two mappings (one applicable to the S-function, one to the data).

The added complexity for the 2nd-order resistance, aside from mappings applied to data external to the modified core cipher, amounts to approximately tripling the number of exclusive-or operations and replacing the fixed S-functions with changeably mapped S-function inputs and outputs, including use of unpredictable data. Externally, data to be operated upon must be replaced by a mapped value and the mapping selection data, and keys must be initially mapped and subsequently incrementally mapped, and stored including the additional mapped data. In the example, the storage requirements for mapped data are tripled. Output data, where this is a key for use in DES, must be stored in this form for future use, except that correlation of the output mappings between output bits and due to relatively static S-function mappings should be removed by incrementally mapping the output using a fresh mapping selection.

It is worth noting that until the S-function is considered, the distinction between use of a blinded data value with mapping selection data and multiple "shares" – equivalent data to be combined using an algorithm to determine the original data – is not clearly apparent. In particular, the operations performed on the mapping selection data up to this point are similar to those performed on the mapped data. However, for the S-functions (and any functions unrelated to or more complex than the mapping operation), it will be seen that the operations applied to the mapping selection data relate to the mapping selection, and only indirectly to the operations of the algorithm. In mapping the input and output of the S-functions, the manipulation of the mapping selection data was quite different from adding similar operations upon "shares" derived from the original data.

ADVANTAGES

The method increases the order of a DPA attack (essentially the number of points in

the observed signal that must be combined to extract any original data). This makes the attack required more sophisticated and complex.

In extending the previously existing concept of blinding to all (or at least more) possible mappings that permit the core operation to remain unchanged, the attacker's task is made more difficult. Where for reasons of economy the selection of mapping used on one data set is related to the selection of mapping for another data set, the larger number of possible mappings might make such a simplification reasonable while not leading to excessive data leakage.

Furthermore, the number of observations needed to extract the original data from noisy side-channel observations may increase substantially more than is achievable through hardware shielding, provided the hardware shielding is high enough. This increase may render even high-order DPA attacks ineffective.

Yet further, the data storage and processing requirements are not increased as much as in some related schemes. An example of such a scheme may be to represent each data bit as a pair of bits, the first value being randomly selected and the second being the original bit when the first bit is zero and its Boolean inverse when it is one (binary "exclusive-or").

Even further, the system containing the cryptographic component can remain unaffected (e.g. protocols can remain unchanged), although cipher choice may be optimised to facilitate use of this technique.

Still further, this technique may be applied with symmetric (having a single, shared secret key) and asymmetric (having distinct but related public and secret keys) ciphers.

In addition, this technique may be applied in conjunction with other techniques for increasing resistance to DPA, such successively modifying the key by use of a complex function for in a co-ordinated fashion with both encryption and decryption.

CLAIMS

1. A method of processing data to reduce the risk of unauthorised access to the data, the method including, in any order, one or more of the steps of:
   (a) inputting a first secret data set into a processor, including possible subsequent repetition of this step;
   (b) providing the processor with a source of unpredictable data;
   (c) providing the processor with a method for selecting suitable mappings by using unpredictable data;
   (d) providing the processor with at least one algorithm for mapping the first secret data set to a mapped form;
   (e) initially mapping the first secret data set, for storage in a mapped form using unpredictably selected mappings;
   (f) modifying an algorithm implementation to operate on mapped and other data including unpredictable data so that the output is mapped data and a mapping selection, to which output the output of the original algorithm operating on the original input data is mathematically related;
   (g) changing the data mapping applied to any data from a prior data mapping by use of a secondary mapping utilising unpredictable information;
   (h) mapping incoming data for input to the modified algorithm implementation; and
   (i) mapping data output from the modified algorithm for further use.

2. A method as claimed in claim 1, wherein the secret data set of step (a) is selected from a cryptographic key and/or a randomisation value.

3. A method as claimed in claim 1 or claim 2, wherein the source of unpredictable data in step (b) is selected from a hardware or pseudo-random number generator and/or stored secret data.

4. A method as claimed in any one of the preceding claims, wherein the data of which the mapping is changed in step (g) is stored, communicated or intermediate calculation data.

5. A method as claimed in any one of the preceding claims, including the step of applying a secondary mapping selected using unpredictable information to the mapped data and/or the mapping on the data.

6. A method as claimed in any one of the preceding claims, including hiding of the initial data mapping on the stored data.

7. A method as claimed in any one of the preceding claims, wherein a data mapping and/or a secondary data mapping are in the form of a lookup-table.

8. A method as claimed in any one of the preceding claims, wherein a data mapping and/or a secondary data mapping are in the form of an algorithm with mapping-selection data or parameters.

9. A method as claimed in any one of the preceding claims, wherein the mapped data and/or the mappings are transmitted to a remote location.

10. A method as claimed in any one of the preceding claims, wherein the mapping is performed by way of a set of mappings selected such that when the input operands are substituted by using such mappings, the original output may be recovered from the resulting output by using a mapping derived from the input mappings.

11. A method as claimed in claim 10 wherein for any operand, the set of all input mappings may be determined from the output mapping by use of a predetermined algorithm.

12. A method as claimed in any one of the preceding claims, wherein the mapping selection is arbitrary, such that each value may be mapped to any value of the range by means of selection of a mapping.

13. A method as claimed in any one of the preceding claims wherein an operation of the original algorithm is modulo-m addition described by $x + y \equiv z \pmod{m}$ and the corresponding operation used in the modified algorithm is to remain the same. The permitted mappings are described by $x_i \equiv a_i x + b_i \pmod{m}$, $y_i \equiv a_i y + c_i \pmod{m}$, and $z_i \equiv a_i z + b_i + c_i \pmod{m}$, in which mappings a_i is any number that is mutually prime with m and b_i and c_i are any numbers so that a_i, b_i and c_i are restricted to the range 0 to $m - 1$.

14. A method as claimed in claim 13, wherein m is of the form $m = 2^n$ so that there are 2^{2n-1} valid selections of mapping for the input x corresponding to all odd values of a_i and all values of b_i and for each of these selections there are a further 2^n valid selections of mapping for the input y corresponding to all values of c_i, with a_i, b_i and c_i restricted to the range 0 to $m-1$.

15. A method as claimed in any one of claims 1 to 12, wherein an operation of the original algorithm corresponds to modulo-m multiplication described by $xy \equiv z \pmod{m}$ and the corresponding operation used in the modified cipher is to remain the same. Permitted mappings are described by $x_i \equiv b_i x^{ai} \pmod{m}$, $y_i \equiv c_i y^{ai} \pmod{m}$, and $z_i \equiv b_i c_i z^{ai} \pmod{m}$ in which mappings a_i is any number mutually prime with $\phi(m)$ which is Euler's Totient function of m, and b_i and c_i are any numbers mutually prime with m so that a_i, b_i and c_i are restricted to the range 0 to $m-1$.

16. A method as claimed in any one of the preceding claims wherein the mapping includes addition of two vectors of n components over the field Z_2.

17. A method as claimed in claim 15, wherein the mapping applied to at least one of the vectors has the form $x_i = A_i x + b_i$ (where lower case indicates a vector and upper case indicates a matrix), wherein A_i is any of $\prod_{k=0}^{n-1}\left(2^n - 2^k\right)$ matrices having an inverse and wherein b_i has any of 2^n values.

18. A method as claimed in claim 17, including mechanisms of selection of A_i and/or b_i whether fixed, restricted or random, wherein the matrices A_i include the identity matrix and bit-permutations of the vector.

19. A method as claimed in any one of the preceding claims, wherein the algorithm includes the use of unary operators.

20. A method as claimed in claim 19, wherein the operator is modified to permit almost arbitrary mappings on the input and output paths and of the operator.

21. A method of processing data to reduce the risk of unauthorised access to the data, the method including data hiding through varying of mapping of some or all data being processed onto a mapped form for computation and/or storage.

22. A computer system or algorithm for processing data to reduce the risk or ease of unauthorised access to the data, the system including one or more of:
    (a) a processor for processing a first data set by means of a first cryptographic key used by the processor; and
    (b) providing the processor with at least one algorithm for mapping at least one of the first data set and the cryptographic key to a mapped form;
    (c) a storage means for storing at least one of the first data set and the cryptographic key in mapped form;
    (d) modifying the algorithm implementation to operate on mapped data;
    (e) periodically changing the data mapping from any prior data mapping by use of a secondary mapping;
    (f) mapping incoming data for input to the modified algorithm implementation; and
    (g) mapping data output from the modified algorithm for further use.

23. A system or algorithm as claimed in claim 22, further including data input means and data output means.

24. A system or algorithm as claimed in claim 22 or claim 23 further including communication means for communicating with a remote computer or terminal.

25. A method as claimed in any of the preceding claims, wherein a mapping selection is determined by any method, including methods of determination that use predictable or unpredictable information, that have uniform and non-uniform probabilities over the possible or valid selections, and have constraints applied to the range of selections available for whatever reason.

Figure 1

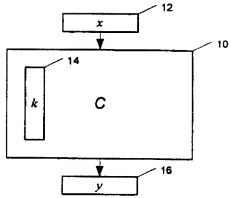


Figure 2

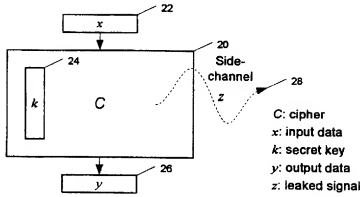


Figure 3

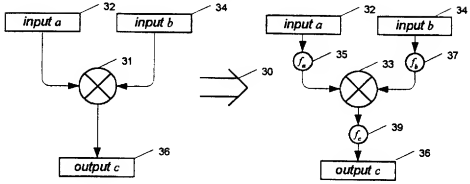


Figure 4

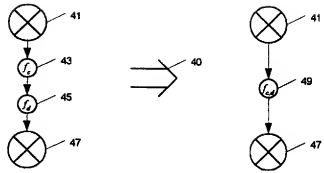


Figure 5

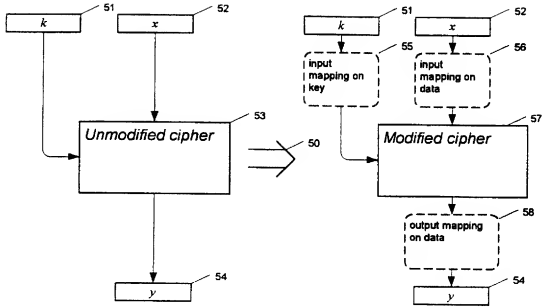


Figure 6

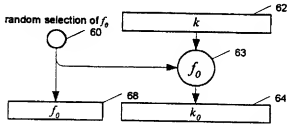


Figure 7

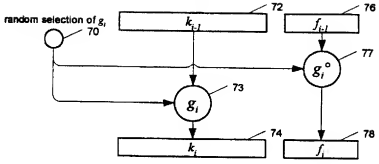


Figure 8

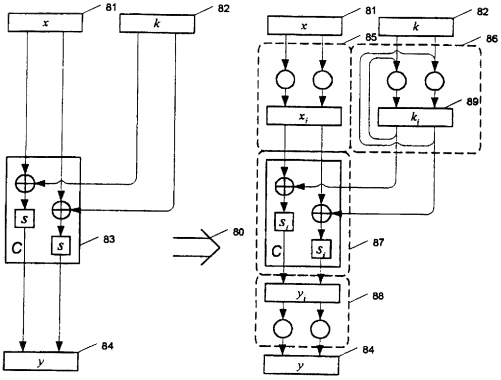
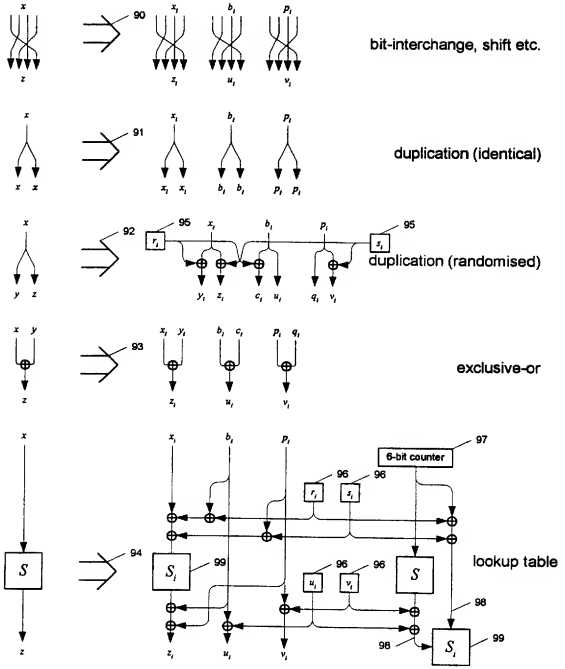


Figure 9


bit-interchange, shift etc.
duplication (identical)
duplication (randomised)
exclusive-or
lookup table
6-bit counter
90
91
92
93
94
95
96
97
98
99